TalkEntreprise

Table des matières

[Introduction 3](#_Toc452888929)

[Description du projet 3](#_Toc452888930)

[Motivation 3](#_Toc452888931)

[Cahier des charges 3](#_Toc452888932)

[Particularité du projet 3](#_Toc452888933)

[Matériel Utilisé 3](#_Toc452888934)

[Logiciels utilissés 3](#_Toc452888935)

[Analyse fonctionnelle 3](#_Toc452888936)

[Préambule 4](#_Toc452888937)

[Point de vu du client/Administrateur 5](#_Toc452888938)

[fonctionnalité spécifique aux administrateur 6](#_Toc452888939)

[Choix des composants 6](#_Toc452888940)

[Schémat de l'application 6](#_Toc452888941)

[Analyse organique 6](#_Toc452888942)

[Choix du type de communication 6](#_Toc452888943)

[Diagramme de classes 6](#_Toc452888944)

[Fichier utile pour l'application 6](#_Toc452888945)

[Du coter Client 6](#_Toc452888946)

[Du coter Serveur 6](#_Toc452888947)

[Méthode d'envoie des messages 6](#_Toc452888948)

[Descritpion des méthodes principal 6](#_Toc452888949)

[Protocole de tests 6](#_Toc452888950)

[Conclusion 6](#_Toc452888951)

[Difficultés rencontrées 7](#_Toc452888952)

[Amélioration future 7](#_Toc452888953)

# Introduction

## Description du projet

Le but de ce projet est de créer une messagerie instantanée pour que les employés d'un même service, puissent communiquer entre eux.

## Motivation

Durant ma formation j'ai appris à utiliser différents langages de programmation, et j'ai choisi de réaliser mon projet en "C Sharp", car j'ai plus de facilité à l'utiliser qu'un autre langage.

Mon choix de créer une messagerie instantanée m'a été inspiré par mes camarades de classe.

Cela remonte à ma première année au CFPT : mes camarades et moi avions parlé de créer un site internet avec lequel nous aurions pu nous envoyer des messages durant les cours.

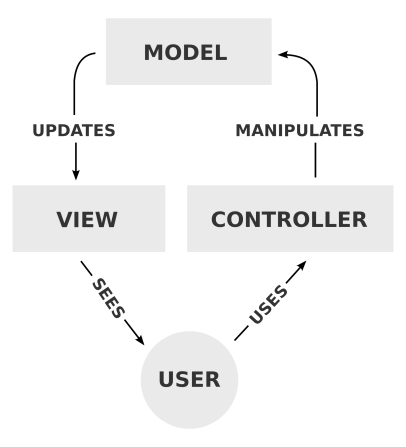
Nous avons essayé, mais nous nous sommes vite rendu compte que nous n'avions pas encore toutes les compétences nécessaires à sa réalisation.

En début d'année scolaire, nous en avons parlé à nouveau. J'ai alors décidé de me lancer ce défi pour mon TPI.

# Cahier des charges

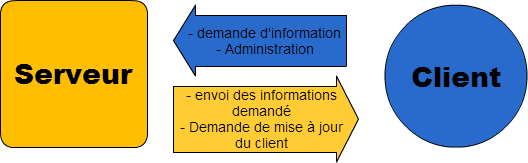
## Particularité du projet

Ce projet va être créé en C Sharp en Windows Form.

* Le programme va être réalisé en MVC (Modèle Vue Contrôleur).

Il y a 2 applications distinctes. Le côté serveur et le côté client

1 Concept du MVC



2 explication des 2 applications

Cette application va permettre d’améliorer la communication des informations dans un même département.

## Matériel Utilisé

* PC windows 7
* Disque dur externe
* Journal de bord (papier)

## Logiciels/fichiers utilissés

* Visual studio 2013
* Easyphp 14.1 VC9
* Suite office 2010
* Github desktop
* Cacoo.com
* mysql-connector-net-6.9.7
* WindowsBase.dll
* MysqlData.dll

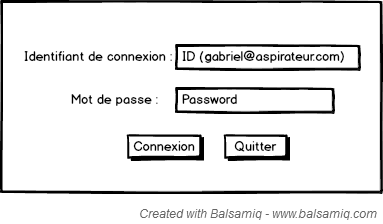
# Analyse fonctionnelle

## Préambule

Pour pouvoir utiliser cette application dans de bonnes conditions, plusieurs interfaces ont dû être créées.

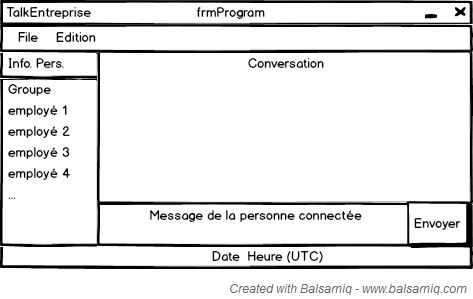
* Application "client"
  + Interface de connexion
  + Interface du programme
  + Interface de déconnexion
  + Interface "à propos"
  + Interface permettant de changer le mot de passe
* Application" serveur"
  + Interface de connexion
  + Interface du programme
  + Interface de déconnexion
  + Interface "à propos"
  + Interface permettant de changer le mot de passe
  + Interface permettant de modifier les informations des utilisateurs

## Point de vue du client/Administrateur



3 Interface de connexion

Cette interface permet à n’importe quel utilisateur de se connecter.



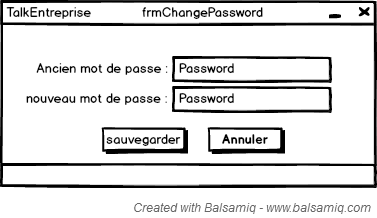
4 Interface du programme

Cette interface permet à l'utilisateur :

* d'envoyer un message
* d'accéder à la fenêtre de modification du

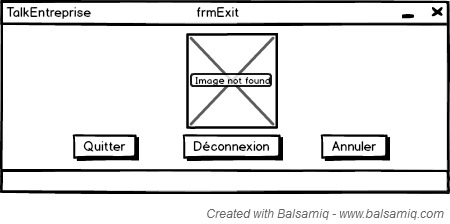
mot de passe

* de consulter la fenêtre "à propos"
* d'accéder à la fenêtre de déconnection.



5 Interface de modification du mot de passe

Cette interface permet à l'utilisateur la modification de son mot de passe.

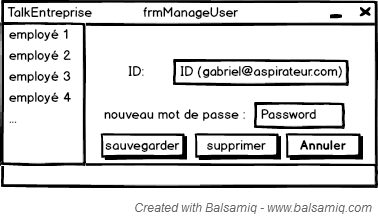


6 Interface de déconnexion

Cette interface permet à l’utilisateur :

* de quitter le programme
* de se déconnecter
* d’annuler la fermeture du programme.

## fonctionnalité spécifique aux administrateur



7 Interface permettant la modification des utilisateurs ainsi que leur suppression

Cette interface permet à l'administrateur de modifier/supprimer un utilisateur. Elle est accessible depuis l'interface du programme (voir image Interface du programme)

## Choix des composants

Mon programme est principalement composé de "textbox". J'ai choisi ce composant car grâce à lui je peux récupérer les informations tapé par l'utilisateur. Et le plus important est l'affichage des différentes conversations. Je peux décider de modifier à tout moment l'affichage des messages et le texte s'adapte automatiquement à la taille du "textbox".

Pour ce qui est de l'affichage des employés, j'ai décidé d'utiliser une "listbox" car elle me permet d' afficher plusieurs informations. Dans mon cas c'est si l'employé est connecté, le nom de celui-ci et le nombre de message en retard.

## Schémat de l'application

# Analyse organique

## Choix du type de communication

Lors de la conception du projet je pensais que l'application client allait chercher les données automatiquement sur la base de données. Mais après avoir discuté longuement avec Monsieur Garcia de la manière de récupérer les données, il m'a conseillé de creuser l'envoie de données via le réseau. En pesant le pour et le contre de nos méthodes, j'en ai conclus que l'envoie de donnés à travers le réseau était la meilleurs solution.

## Diagramme de classes de l'application CLient

## Diagramme de classes de l'application SERVEUR

## Fichier utile pour l'application

### Du coter Client

#### Les vues :

#### Les classes :

### Du coter Serveur

#### Les vues :

#### Les classes :

## Méthode d'envoi des messages

## Descritpion des méthodes principal

# Protocole de tests

# Conclusion

## Difficultés rencontrées

Au cours de ce projet, j'ai rencontré certaines difficultés qui de première abord étaient simple ou complexe. Cela étant, je vais vous expliquer les problèmes qui étaient pour moi le plus compliqué

Ma première difficulté est apparue lors de notre conversation avec Monsieur Garcia au sujet du moyen de communiquer les informations de la base de données aux utilisateurs. En réfléchissant j'ai opté pour l'envoi de données via le réseau. En choisissant cette option, Monsieur Garcia m'a donné une piste. J'ai alors dû rechercher sur le web des informations relatif à la méthode de communication. Après d'innombrables exemples incompréhensibles et de théorie que je ne comprenais pas. Je suis tombé sur un site qui proposait une démonstration et une explication qu'y m'ont permis de comprendre comment cela fonctionne.

Lors de la création de mon application client, j'ai décidé de lancer la fenêtre principale dans un nouveau processus. Je me suis rappeler d'un ancien programme ou j'avais essayé d'utiliser les threads. J'ai compris que sa permettait de rendre indépendant une partie du code. En revanche je ne savais comment instancier une nouvelle fenêtre dans un nouveau processus. Pendant mes recherches, je suis tombé sur un le bout de code qu'il me fallait. En l'intégrant dans mon application j'ai vu qu'une erreur s'affichait. Sachant que ce genre d'erreur est habituellement dû à un "using" manquant, je l'ai donc ajouté. À mon grand étonnement visual studio disait que le "using" utilisé n'existait pas. J'ai alors pensé que je devais référencer dans ma solution un fichier dll. Et effectivement je devais référencer le dll nommé WindowsBase.dll.

Durant la création de la méthode d’affichage des employés, je n’arrivais pas à donner à ma "listBox" la liste d'utilisateur créées dans un processus à part. L'erreur disait qu'il n'était pas possible de récupérer l'information car elle provenait d'un autre processus. C'est comme ceci que j'ai compris que la commande "Invoke" ne permet pas que d'appeler des méthodes dites encapsuler mais permet aussi de passer des informations aux objets présent sur la forme.

Après avoir réalisé la méthode de récupération et d'affichage des conversations entre 2 utilisateur, je me suis penché sur l'envoie de message dans un group. Tous se passaient bien jusqu'au moment où je remarque que des messages s'affichaient en double. En visionnant étape par étape la création de l'affichage, j'ai remarqué que la requête qui allait chercher les informations dans la base de données me retournait les doublons. C'est là que j'ai compris que lors de l'enregistrement des messages dans la base de données, les messages ne s'enregistrement pas au même moment[[1]](#footnote-1). Pour pallier à ce problème, j'ai décidé de créer une seconde méthode qui enregistre d'abord les informations dans la base de données et ensuite mettre à jour la vue des clients.

## Amélioration future

1. Message : "bonjour"🡪enregistrement base de données:

   user1;"bonjour";2016-02-03 12:02:00

   user2;"bonjour";Date-12-02 12:02:00 [↑](#footnote-ref-1)